Question: 2. Suppose there are n cities connected with each other via part…

(1 bookmark)

2. Suppose there are n cities connected with each other via particular road. Each road has a length w. Exactly in one city m there is a tiger. Mayor of other cities want to capture that tiger. Mayor who comes first in m city will capture the tiger. Your task is to write a code that finds which city mayor will capture the tiger first.
You are given number of city n, number of road r, length of each road w and the location of the city m that holds the tiger as input. Input will be given in following format:

n
r
u v w

Here, u v w denotes the road between city u and v, length of that road is w. See sample input and output for better understanding.

Sample input	Sample output
n: 4 r: 4 0 1 7 2 3 5 0 3 10 2 1 1 m: 2	Mayor in city 1 will capture first.

Show transcribed image text

Expert Answer

Anonymous answered this
108 answers

Was this answer helpful? 2 1

It is asked to find out which city is the nearest from the source (m).

We can use Dijkstra algorithm to find out the shortest distances from source for all vertices.

And the shortest of all is picked as the result.

```
#include <stdio.h>
#include <limits.h>
#include <stdbool.h>


// function to print the final result
void print(int n, int src, int dist[])
{


	int min, ind, i;
	min = INT_MAX ;

	for (i = 0; i < n; i++) {

		if((dist[i] < min) && (i != src)) {
			min = dist[i];
			ind = i;


		}


	}


	printf("\nMayor in City %d captures first\n", ind);


}

// function to find minimum distance
int mD(int n, int dt[], bool s[])
{

	int mn = INT_MAX, m_in;

	for (int i = 0; i < n; i++)
		if (dt[i] <= mn && s[i] == false) {
			mn = dt[i];
			m_in = i;

		}


	return m_in;
}


// dijkstra algorithm to find shortest paths from the source each of the vertex.

void Dij(int n, int G[n][n], int src)
{
	int dt[n];

	bool s[n];


	for (int i = 0; i < n; i++) {
		s[i] = false;

		dt[i] = INT_MAX;

	}


	dt[src] = 0;


	for (int c = 0; c < (n-1) ; ++c) {

		int z = mD(n, dt, s);


		s[z] = true;


		for (int v = 0; v < n; v++)


			if !s[v] && G[z][v] && dt[z] != INT_MAX
				&& dt[z] + G[z][v] < dt[v])
				dt[v] = dt[z] + G[z][v];
	}


	print(n, src, dt);
}


int main()		// main function
{

	int n, r, m, val, ind_end, ind;
	printf("\nn:");
	scanf("%d", &n);	// console input for n
	printf("r:");
	scanf("%d", &r);	// console input for r

	int edg[r][3];	// input for edges
	for (int i=0; i<r; i++) {
		scanf("%d %d %d", &edg[i][0], &edg[i][1], &edg[i][2]);

	}

	printf("m:");
	scanf("%d", &m);	// input for source


	int graph[n][n];

	for(int i=0; i<n; i++) {

		for(int j=0; j<n; j++) {

			graph[i][j] = 0;
		}
	}


	for(int i=0; i<r; i++) {		// preparing adjacency matrix from the given input
		ind = edg[i][0];
		ind_end = edg[i][1];
		val = edg[i][2];
		graph[ind][ind_end] = val;
		graph[ind_end][ind] = val;

	}

	Dij(n, graph, m);  // calling Djikstra algorithm with source m

	return 0;
}
```

Output:

```
4
4
0 1 7
2 3 5
0 3 10
2 1 1
2

Mayor in City 1 captures first
```

View comments (1)

Questions viewed by other students

Q: 1. Suppose a team is working to develop an artificial agent to detect the fish in underwater. Using camera and others sensors, task of the agent is to identify the different types of the fish in water. Explain with proper reasons which of the following should the suitable environment for the agent. a. Fully observable vs partially observable b. Deterministic vs stochastic c...

A: See answer